7-21-2010

Combining Dynamic Modeling With Geometric Constraint Management to Support Low Clearance Virtual Manual Assembly

Abhishek Seth
Caterpillar Inc.

Judy M. Vance
Iowa State University, jmvance@iastate.edu

James H. Oliver
Iowa State University, oliver@iastate.edu

Combining Dynamic Modeling With Geometric Constraint Management to Support Low Clearance Virtual Manual Assembly

Abhishek Seth
Department of Applied Research,
Product Development & Global Technology,
Caterpillar Inc.,
Peoria, IL 61629
e-mail: seth_abhishek@cat.com

Judy M. Vance
Fellow ASME
e-mail: jmvance@iastate.edu

James H. Oliver
Fellow ASME
e-mail: oliver@iastate.edu

Department of Mechanical Engineering,
Virtual Reality Applications Center,
Iowa State University,
2274 Howe Hall,
Room 1620,
Ames, IA 50011

This research presents a novel approach to virtual assembly that combines dynamic modeling with geometric constraint-based modeling to support low clearance manual assembly of CAD models. This is made possible by utilizing the boundary representation solid model data available in most contemporary CAD representations, which enables (a) accurate collision/physics calculations on exact model definitions, and (b) access to geometric features. Application of geometric constraints during run-time, aid the designer during assembly of the virtual models. The feasibility of the approach is demonstrated using a pin and hole assembly example. Results that demonstrate the method give the user the ability to assemble parts without requiring extensive CAD preprocessing and without over constraining the user to arrive at predetermined final part orientations. Assembly is successful with diametral clearance as low as 0.0001 mm, as measured between a 26 mm diameter hole and pin. [DOI: 10.1115/1.4001565]

Keywords: virtual reality, virtual prototyping, human-computer interaction, virtual assembly, constraint-based modeling, physical constraint simulation

1 Introduction

Assembly processes constitute a majority of the cost of a product [1], thus it is crucial to establish a comprehensive assembly planning process that can easily identify actual assembly issues such as assembly sequences, ergonomics, and operator safety. A well-designed assembly process can improve efficiency and quality, reduce cost and time to market. Computer-aided assembly process planning typically focuses on developing algorithms to automatically generate assembly sequences. These methods generally require extensive expert knowledge, which often limits their effectiveness. Commercial CAD programs allow designers to generate geometric constraint relationships among models to develop assembly simulations. Once created, these assembly sequences can be recorded and visualized as 3D simulations.

However, these methods do not account for the effect of human interactions involved during assembly. Currently, designers must interact with complex 3D CAD models using two-dimensional devices like a mouse and a keyboard while viewing the models using a flat computer screen. While such interfaces allow designers to verify geometric interferences and other aspects in assemblies, their two-dimensional nature make it difficult to predict issues that arise when an assembly worker is instructed to assemble the parts for the first time. For example, the restrictive nature of the two-dimensional interface does not allow direct 3D manipulation of parts to simulate how humans will interact (reach out, grab and manipulate) with complex models during assembly. Human modeling software such as VISJACK [2] are available and often used to prototype assembly methods, however, VISJACK remains a simulation of human interaction that represents simulation of the human body. In this way, the software once again models how a human would complete the assembly process and it does not provide direct interaction with virtual product models. The result is that many problems with the assembly process are found later in the product design process, perhaps even on the assembly line, when the first physical prototypes are built.

Virtual reality technology offers a solution to this problem by providing a three-dimensional immersive environment in which users can interact using natural human motions. Virtual reality technology enables human-computer interaction through the stimulation of multiple senses, including the visual, haptic, and auditory senses, to immerse the user in a computer-generated world. Developing virtual reality simulations for manual assembly is difficult due to the need to simulate the continuous and subtle human interactions that are involved. Other challenges include handling large and complex CAD data sets and real time simulation of physical constraints.

2 Challenges and Related Work

The challenges to developing a virtual environment to enable manual virtual assembly are many. In virtual assembly, users are manipulating complex CAD geometry that contains convex and nonconvex surfaces. Collision detection and force calculations between complex CAD models provide feedback to the user during simulated manual assembly. Accurate modeling of the interaction of mating parts is essential to advance the use of manual virtual assembly as a prototype and design tool. The focus of this research is to develop methods that allow human-in-the-loop evaluation of the assemble-ability of parts and the determination of a feasible assembly sequence.

2.1 Mechanical Assembly: Human in the Loop. In order to illustrate the challenges of manual virtual assembly, a simple assembly task of inserting a pin into a hole is chosen. Figure 1 shows two parts: a pin part and a block with a hole. The pin

Contributed by the Design Theory and Methodology Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received May 16, 2007; final manuscript received March 24, 2010; published online July 21, 2010. Assoc. Editor: John K. Gershenson.

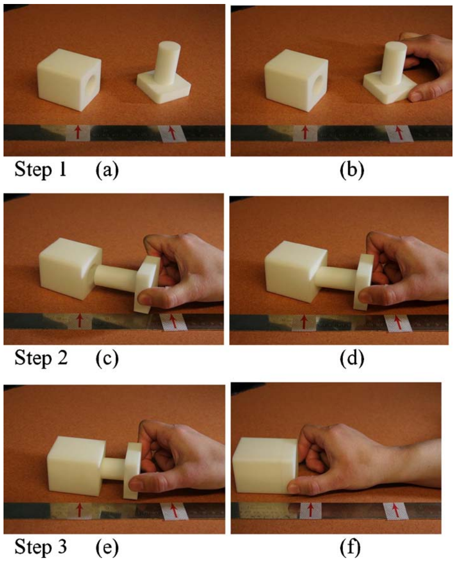

Fig. 1 Assembly sequence of pin and hole: (*a* and *b*) step 1, (*c* and *d*) step 2, and (*e* and *f*) step 3

diameter is 25 mm and the hole diameter is 26 mm. The manual assembly task can be divided into three separate steps (refer to Fig. 1).

- Step 1: (a) Approach the worktable on which the two parts are placed and (b) grasp the pin.
- Step 2: (c) Manipulate the pin and (d) align it roughly with the hole.
- Step 3: (e) When aligned, (f) push the pin into the hole to complete the assembly.

Simulating simple assembly tasks, such as the pin-in-the-hole assembly, in a virtual environment presents several challenges. By analyzing the steps in detail, it is evident that to accomplish the first step, the simulation should allow the user to interactively select any part present in the environment.

Collision detection is frequently used as a method to select parts in a virtual scene. A virtual hand model is constructed to represent the user's hand into the computer-generated environment. Position trackers are used to register the movement of the user's hand with the virtual hand model. Collisions are detected between the virtual hand model and other complex part models present in the environment. Once the hand model collides with the part model, the user presses a button or makes a gesture to grab the colliding part. The motion of the part model then follows the motion of the hand model as if the hand were attached to the part. High accuracy collision detection is not critical to this step.

After the user grabs the part, the second step is to simulate realistic part manipulation in the virtual environment. This requires modeling complex hand-part interactions that will allow the user to rotate and translate the virtual part as they would in the real world. Different grasping techniques are explored by researchers to allow for dexterous manipulation of virtual parts [3,4].

During the third step, as the user inserts the pin into the hole, he/she should feel the collision force exerted by the part interaction, followed by a friction force during insertion. Consider the "hole" part to be freely resting on the table and the pin roughly aligned with the hole. To assemble, the pin will go into the hole until the convex cylindrical surface of the pin collides with the concave cylindrical surface of the hole (Fig. 1(*d*)). The force induced by a slight misalignment will overcome the static frictional force of the freely resting hole part and it will move to align itself to facilitate insertion (Fig. 1(*e*)). Assuming the frictional force of the aligned cylinders is less than that generated by the partial assembly on the table, insertion can be completed. It is evident from the ruler markings in Fig. 1(*f*) that once the pin is completely inserted into the hole, the user can push the entire assembly. Alternatively, if the hole part is held in a fixture, once the cylindrical surfaces collide and the user pushes the pin, the hole part will exert an appropriate reaction force on the pin that will be felt by the user. This force helps align the pin properly to facilitate assembly. In this step, accurate collision detection and force calculations are critical to achieving a realistic simulation.

Simple assembly tasks like inserting a pin into a hole consist of complex interactions that require depth perception for grabbing and proper alignment, precise part manipulation, haptic perception, and realistic part behavior. Simulating such behavior requires the system to be capable of detecting collisions between the pin and the hole surfaces with very high accuracy. Once collisions are detected, physical responses must be modeled to reproduce realistic behavior of the rigid bodies. It is important to note that the pin and hole assembly described above provides a simple-to-understand but challenging example for collision detection and physics-modeling because of the low clearance nonconvex nature of the geometry involved.

2.2 Background. Current methods of simulating virtual manual assembly tasks use a combination of part snapping, physics-based modeling, and geometric constraint modeling. Part snapping relies on using predetermined final part assembly positions. As one part approaches the vicinity of a mating part, it snaps into its final position. Ritchie and co-workers [5–7] used part snapping techniques for virtual manual assembly simulation with the goal of generating assembly plans. The authors state that the lack of hardware and software to simulate exact positioning of parts led them to implement part snapping [6]. It is important to note that part snapping methods do not take into account any physical interaction (part contacts, acceleration, colliding and gravitational forces, etc.) while simulating assembly.

Researchers have attempted to model the physical behavior of parts in virtual environments to facilitate realistic interaction and dynamic response for assembly tasks. A desktop-based system developed by Gupta and co-workers [8,9] applied physics-based modeling but is limited to 2D models for assembly. Coutee et al. [10,11] employed a similar desktop-based dual hand system and relies on collision detection and physics computations for assembly. This system, called HIDRA, is limited to handling nonconvex CAD geometry, and thus, is only suitable for simulating assembly operations among simple primitive-based models. Fröhlich et al. [3] used the CORIOLIS™ [12] physics-based simulation package and the Responsive Workbench [13] for simulating bench assembly scenarios in virtual environments. For this system, interactive update rates are difficult to maintain when several hundred collisions occurred simultaneously and at least 5% clearance is necessary to avoid numerical instabilities. Mandiak and Kesavadas [14] illustrate their virtual manual assembly method with a pin-in-a-hole assembly task where the interaction forces are based on methods derived by Whitney [15]. Kim and Vance [16,17] used the VoxMap PointShell method [18] to support the physics-based modeling of virtual manual assembly tasks.

An alternate approach to enable virtual assembly simulation relies on utilizing interpart geometric constraints for guiding assembly instead of snapping parts to a predefined assembly position. Once the constraints are defined among the parts, a geometric constraint solver calculates the new (generally fewer) degrees-of-freedom to enable precise relative positioning of parts, thus simplifying assembly. Jayaram and co-workers [19–23] created VADE, which uses Pro/Toolkit to import assembly data (transformation matrices, geometric constraints, assembly hierarchy etc.) for simulating assembly operations in a virtual environment. Predefined geometric constraints are activated to simulate constrained motion when parts approach mutual proximity. Parts are then snapped to their final position to complete the assembly task.

Wang et al. [24] implemented a physics-based algorithm with limited capabilities into VADE to simulate a more realistic part behavior. A constraint manager that was developed by Marcelino et al. [25] enabled interactive simulation of assembly/disassembly. In this paper, simple planar and cylindrical surfaces are used for defining and validating constraints, and solving constrained motion. A CAVE-based system for virtual assembly called MIVAS was presented by Wan et al. [26]. Similar to VADE, MIVAS used Pro/Toolkit for importing CAD geometry and predefined geometric constraints from Pro/Engineer CAD software. Liu and Tan [27] used constraint-based modeling for assembly and tolerance analysis. Different constraint criteria (proximity, orientation, etc.) were applied among parts. Chen et al. [28] described a system called VECA, to allow engineers to perform collaborative assembly tasks. Similar to VADE and MIVAS, VECA also used Pro/Toolkit for extracting geometry and constraint data from the Pro/Engineer CAD software.

2.3 Motivation. Prior research reveals that part snapping, physics-based, and geometric constraint-based methodologies have limitations when used to simulate assembly tasks. Modeling large and complex assemblies consisting of nonconvex CAD models with large polygon counts present challenges for assembly simulation when accurate and timely modeling collision and physics responses are critical [17,29]. When simulating assembly tasks with physics-based methods (even those as simple as the pin and hole assembly), several hundreds or thousands of polygon collisions occur simultaneously among the mating parts, resulting in numerical instabilities and making simulations noninteractive [3,10,11,30]. Volumetric representations [18,31] derived from polygon-based CAD models can be used to speed up the calculations, but these methods sacrifice accuracy by using coarser geometric representations, which do not allow CAD parts to be assembled with clearances typical in mechanical design [3,4]. In addition, most virtual assembly applications that use geometric constraint-based methods rely on importing specific metadata (transformation matrices, geometric constraints, assembly hierarchy, etc.), which results in time consuming and cumbersome preprocessing of the geometry. Because constraints are preimported, these systems do not allow users to change assembly relationships within the virtual environment. It is important to note that geometric constraint methods do not take into account any physical interaction (part contacts, acceleration, colliding and gravitational forces, etc.) while simulating assembly.

When clearance between parts is small, precise movement and alignment is required to complete the assembly task. Current VR hardware (trackers and 3D input devices) lacks the accuracy necessary to perform precise manipulation of parts in the virtual space. In practice, the inaccuracies associated with the input hardware causes unnecessary collisions among objects when trying to perform low clearance assembly tasks. Thus, not only is it important to develop methods to accurately model physical constraints, methods are needed to overcome hardware limitations of the virtual environment to enable precise part movement and alignment necessary to complete the assembly task.

Clearly, it is challenging to successfully simulate all aspects of low clearance assembly of complex, nonconvex CAD models using the approaches described previously. Approaches focusing on interactively simulating physical constraints among part surfaces provide the advantage of building an environment that more accurately simulates real world dynamics of manual assembly tasks by including the human as an integral part of the process. Geometric constraint-based methods, on the other hand, allow precise manipulation of part models to complete assembly with minimal computation load.

The research presented here proposes a new method, which combines the advantages of physics-based modeling with those of geometric constraint-based modeling. The long term goal is to provide an immersive, haptically enabled manual part assembly methodology.

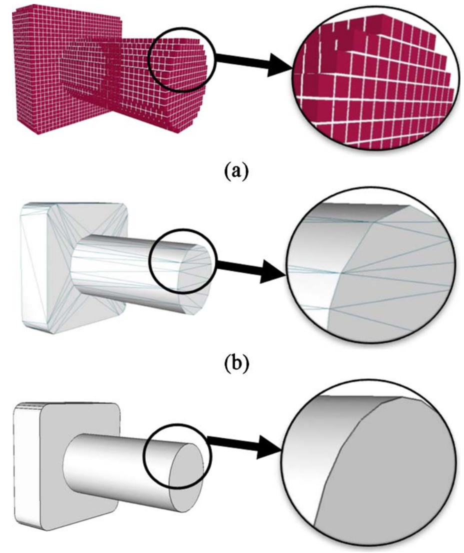

Fig. 2 (*a*) Voxel, (*b*) polygon, and (*c*) B-Rep representations of a part

3 Methodology

Performing low clearance assembly using nonconvex CAD models within virtual environments presents two critical modeling requirements:

1. highly accurate collision detection and physics-based dynamic modeling among contacting geometric surfaces; and
2. precise part manipulation and alignment of geometric mating features.

This method takes advantage of physics-based methods to simulate dynamic behavior of colliding parts and geometric constraint modeling to support precise part alignment and manipulation.

3.1 B-Rep Solid Model Data. In most previous virtual assembly research, the input CAD models, consisting originally of individual precise parametric surface and/or B-Rep solid models, are translated into polygonal model representations for graphics visualization within virtual reality environments. This polygonal data also is also typically used for collision and physics-modeling calculations [4,5,10,17,19,26]. Figure 2 shows a voxelized model, a polygonal model, and a B-Rep model of the same part.

The B-Rep solid model is a representation of the boundary between solid and nonsolid, as defined by a set of connected surface elements. In this research, topological and geometry information such as faces, edges, and vertices of the B-Rep solid model are used to indicate surface collisions and identify geometric constraints. The Parasolid® (x_t) industry CAD standard provides lossless B-Rep data for manipulating in the virtual environment.

The use of B-Rep solid models has the following advantages.

1. It facilitates highly accurate collision detection and physical constraint simulation, necessary to achieve low clearance assembly.
2. Provides access to geometric feature information (surface topology, faces, edges, etc.) that allow users to create/delete geometric constraints manually within the virtual environment.

3.2 Collision Detection and Physics-Based Dynamic Modeling. Collision detection is used for selecting parts in the environment and for detecting interpart collisions. The use of B-Rep solid models for computing collisions among parts allows the system to compute results with a much higher accuracy as the solver detects contacts between exact surface geometries instead of faceted approximations. This is a major improvement to existing virtual assembly methods, which operate on approximate geometric models. Physics-based dynamic modeling, to simulate part motion during collision, is also based on the B-Rep solid model data.

3.3 Precise Part Alignment and Manipulation. Geometric constraint-based methods are used to address the challenge of precise part alignment and manipulation. These constraints are used in special cases to augment physical constraints for facilitating the assembly operation. Geometric constraints provide specific advantage during low clearance assembly by reducing a part's degree of freedom, thus allowing the user to precisely manipulate and align parts in the virtual environment. Once colliding features of two mating parts are detected, the B-Rep solid model data is used to define constraint relationships between these features. These geometric constraints are used to augment physical constraint modeling and are applied during run-time for facilitating the assembly operation. Geometric constraints do not need to be defined by preprocessing CAD data, but are created during run-time when two appropriate surfaces collide. Once geometric constraints are defined, the solver takes into account both physical and geometric constraints for computing part trajectories where geometric constraints ensure precise alignment and physical constraints ensure dynamic behavior. The defined geometric constraints can be deleted at any time by the user by voice or menu command. Access to lossless B-Rep data representation and feature information within the virtual environment supports this methodology. The use of a standard CAD format for importing B-Rep solid models eliminates the need for proprietary software toolkits and allows geometry from multiple CAD systems to be imported.

Previous attempts [19,26–28] using constraint methods were prone to the following different limitations.

(1) Geometric constraints had to be predefined and imported from a specific CAD system before assembly could be performed.
(2) Specific CAD toolkits were required to access the proprietary CAD metadata (constraints, assembly hierarchy etc.) from the CAD system.
(3) In addition, the geometric constraint relationships could not be defined or modified interactively within the virtual environment.

This research is one of the first attempts to successfully demonstrate a combination of physics-based and constraint-based behavior for virtual assembly where both physical and geometric constraints are created and deleted at run-time.

4 Implementation

The methodology was implemented within the SHARP software [32]. SHARP consists of three core components, namely, the application platform, the visualization engine, and the physics engine, as shown in Fig. 3. SHARP utilizes the VRJUGGLER [33] software toolkit for controlling the virtual environment. OpenSceneGraph, an open-source scene graph library is used for visualization. The 3D input devices include two PHANTOM Omnis® and the tracking system is a Polhemus PATRIOT™.

The D-Cubed family of software components from Siemens® are used for collision detection, physics, and constraint behavior simulation in the virtual environment. The collision detection manager (CDM) is used by the collision module for calculating and querying collision/interference information, and the dimensional constraint manager (DCM) [34] is used by the constraint module for defining and solving for geometric constraints. The assembly engineering manager (AEM) is used by the physics module for manipulating solid parts in the virtual environment. AEM integrates mass and inertia properties to the geometry model

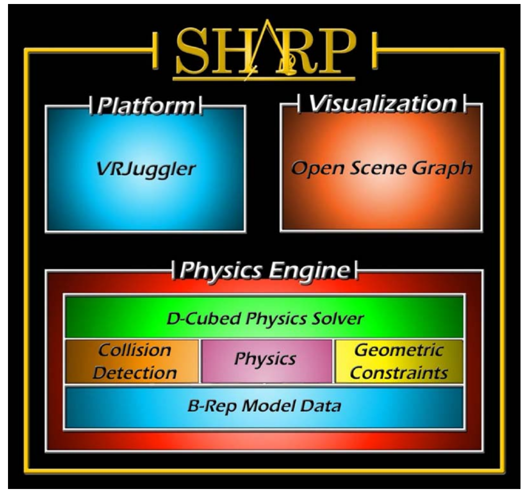

Fig. 3 SHARP modular structure

for simulating part contacts and forces, acceleration, etc. during the simulation.

During the initialization step, the application sets the linear resolution, which determines the accuracy at which computations are performed by D-Cubed. A resolution setting of 0.01 mm indicates that the system will identify a contact when distance between two geometries is less than or equal to 0.01 mm. Further reducing this resolution setting will result in a more accurate solution with an associated increase in computational time.

Figure 4 shows the SHARP application flowchart. The application first reads a configuration file, which contains data about the initial assembly environment setup such as number of parts, initial positions, etc. Once B-Rep and graphic data models are loaded, the user can reach and grab models in the virtual environment and start the assembly process. The system relies on collision detection for selecting parts in the scene. Once a part is selected by the user, an AEM-based physics sequence is initiated. This allows the user to manipulate the model, move it freely in space, and place it in its final desired position. The system detects collisions between the models present in the scene and allows the user to guide the part into position while experiencing the effect of physics-based modeling of part interactions. Collision detection and physics simulation allow the user to collide parts together, push other parts realistically, and experience the effect of gravitational and interaction forces. When low clearance assembly scenarios are encountered, users interactively identify mating features with the input device and use voice commands to apply appropriate geometric constraints to achieve precise part manipulation to facilitate the assembly task. All physics computations are performed in a separate high-priority thread to maximize the use of available CPU performance.

5 Example

The pin and hole assembly example as described in Sec. 2 is used to test the proposed methodology. The virtual pin and hole are modeled with the same dimensions as the ones used in the real world assembly demonstration (Fig. 1), resulting in a pin diameter of 25 mm and a hole diameter of 26 mm. A linear resolution setting of 0.01 mm in D-Cubed is used for this paper. In order to illustrate the functionality and capabilities of the proposed method, four examples are presented: collision detection only, dynamic modeling only, collision detection and dynamic modeling, and finally, the proposed method, which includes collision detection, dynamic modeling, and geometric constraints.

5.1 Case I: Collision Detection Only. In this example, only collision detection is available to assist the user during assembly.

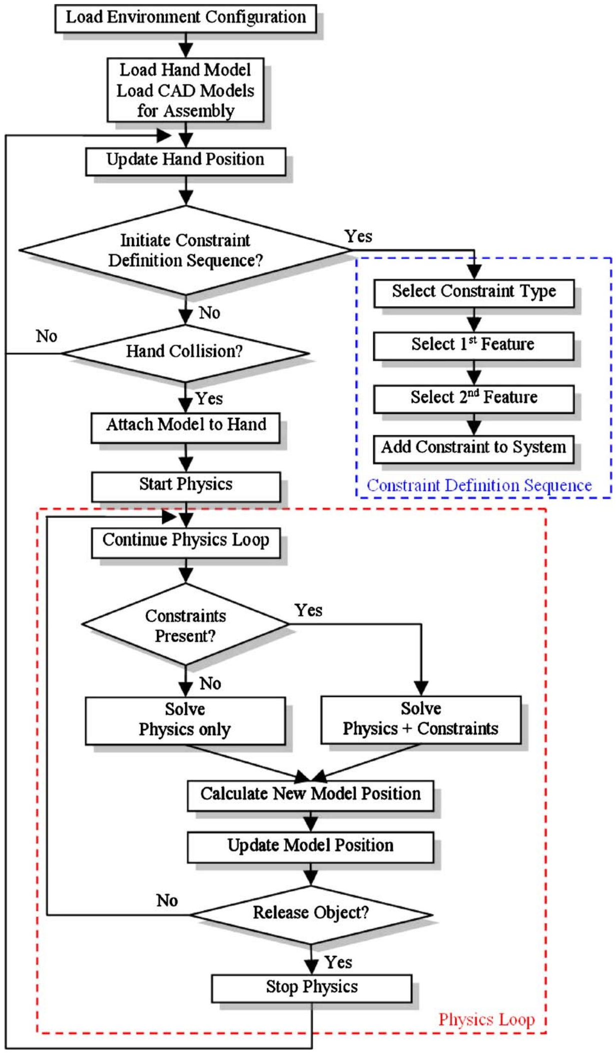

Fig. 4 Application flowchart

Collisions are detected among models only to prevent interpenetration. The user picks up the pin part and aligns the pin with the hole. While inserting the pin into the hole, the pin stops as soon as it collides with the hole part (Fig. 5). In this case, the system does not provide any intuitive help to the user to facilitate assembly, e.g., there is no physical "self-aligning" response of the hole part as the forces acting during assembly are not modeled. All parts are inherently stationary so the user must align the pin precisely to complete the assembly, which is extremely difficult with the precision of today's interface hardware.

5.2 Case II: Geometric Constraints Only. In this example, constraint-based modeling is used for assembling components. During the first step, the user manipulates and roughly aligns the model (Fig. 6 (a) and Fig. 6 (b)). Then the user starts the constraint definition sequence in which he/she selects the cylindrical

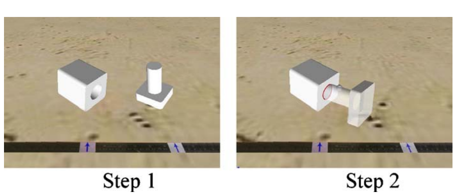

Fig. 5 Assembly using collision detection only

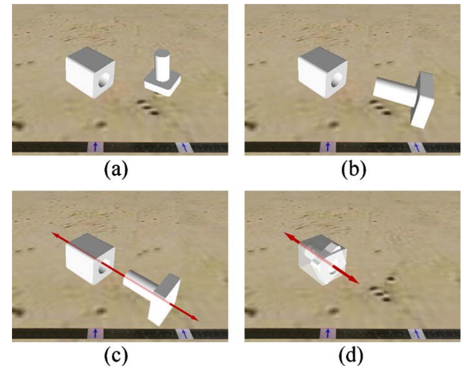

Fig. 6 Assembly using geometric constraints

surface of the hole, then the cylindrical surface of the pin using the mouse or 3D input device. Next, the user instructs the application, through voice commands, to apply a concentric constraint between these two surfaces. The part positions are updated, such that the pin and hole are properly aligned with each other (Fig. 6 (c)). Red arrows passing through the models in Fig. 6 depict the location of the concentric constraint acting between the models. Using the voice interaction module, users define, apply, and delete geometric constraints on-the-fly, as well as launch other system commands.

Applying geometric constraints results in reducing the degrees-of-freedom of the pin part, such that it can only move in and out of the hole and rotate about its axis. Without the presence of collision detection (among the parts), the parts can interpenetrate each other, making the simulation unrealistic (Fig. 6 (d)). No physical behavior among parts (such as the pin pushing the block along the table) is simulated.

5.3 Case III: Collision Detection and Dynamic Modeling. In this example, collisions are detected and part movements are calculated based on the interaction forces between models. When the user tries to insert the pin into the hole, physical constraints (among the colliding surfaces), which are represented by dynamic modeling, facilitate in guiding the pin. Once the end of the pin part enters the hole, interaction forces move the hole part, such that part surfaces are aligned to facilitate assembly. This behavior is similar to what we observed while performing assembly in the real environment.

In this case, however, although collision and physics calculations are very accurate, the 3D input devices and trackers are not precise enough to support the very fine movements required for this low clearance assembly. In addition, once the pin is halfway inside the hole, multiple collisions create an excessive load on the physics solver and slow down the application, causing lag between the user movement and the graphics update.

5.4 Case IV: Collision Detection, Dynamic Modeling, and Run-Time Geometric Constraints. In this example, the new method, which combines collision detection, dynamic modeling, and run-time geometric constraints is used to assemble the parts. The user reaches and grabs the pin part (using collision detection) and roughly aligns it with the hole part (Figs. 7(*a*)–7(*c*)). When the pin and hole parts are close, the user starts a concentric constraint definition sequence (Fig. 7(*d*)) by identifying the constraint surfaces and using voice commands to apply the constraint. Once a constraint is defined and applied, the solver allows the user to move the pin into the hole smoothly (Fig. 7(*e*)). When fully inserted, collisions are automatically detected between the flat face of the pin part and the hole part, preventing part interpenetration. It is important to note that when the user keeps applying force on the pin part, the system calculates the interaction forces at the colliding surfaces and simulates physical responses resulting in the pin part pushing the entire assembly along the surface of the

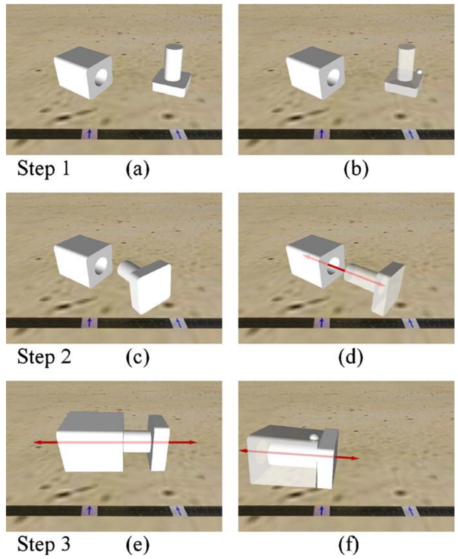

Fig. 7 Assembly using collision detection, dynamic modeling and geometric constraints: (*a* and *b*) step 1, (*c* and *d*) step 2, and (*e* and *f*) step 3

table (Fig. 7(*f*)). Thus, geometric constraints in this case facilitate the assembly task by ensuring precise alignment between parts, while dynamic modeling simulates realistic part movement.

Further testing of the proposed method was done to evaluate the effect of smaller clearances. The results demonstrated that the method allowed pin-hole assembly with diametral clearances as low as 0.0001 mm on a hole with a 26 mm diameter to be completed.

6 Conclusions and Future Work

The paper proposes a new approach to manual assembly simulation in a virtual environment, which combines dynamic modeling and geometric constraint-based modeling to facilitate low clearance assembly of CAD models. The method relies on the use of B-Rep geometry for computations, thereby providing exact collision detection and identification of constraint entities. Geometric constraints are identified and applied at run-time, not during preprocessing. Further, predefined final positions of assembled parts are not required. This method allows the user full freedom to assemble parts in any assembly order or arrangement. The combination of dynamic modeling and geometric constraints enables the user to easily manipulate the parts (supported by dynamic modeling) and guide them together, even in the presence of low clearances (supported by geometric constraints). Case studies are presented using a pin-in-a-hole assembly example, which compares the observed part behavior using previous methodologies to the new approach. This new method supports more natural interaction during the assembly process than either dynamic modeling or geometric constraint method, when used independently.

While the case studies demonstrate the success of using this method to support low clearance assembly, there are still critical issues to overcome before a general purpose algorithm is developed. One issue encountered when implementing this method was the inability of the computational engine to calculate collisions of the B-Reps in time frames short enough to support haptic interaction. Future work will examine methods to improve the computational speed in order to support haptics interaction. In addition, automatic geometric constraint recognition methods are needed, which will enable the system to automatically define the necessary geometric constraints based on the predicted assembly intent of the user. In the future, geometric constraints will be added and deleted automatically during run-time, resulting in more intuitive interaction with the environment. Finally, part fixturing to hold one part stationary as the other part is assembled onto it, as well as functionality to support creation of subassemblies is also in future plans.

In conclusion, this paper presents a new method to support low clearance virtual manual assembly. Until we can support this assembly action, we cannot develop a truly general purpose virtual assembly system in which to perform human-in-the-loop assembly evaluations. Once achieved, though, the ability to perform virtual manual assembly will support a wide range of capabilities involving CAD model manipulation, including design for assembly, design for maintainability, assembly training, repair training, and many others. Achieving low clearance assembly is the key to moving virtual assembly from the research lab into production.

Acknowledgment

We are grateful for the technical assistance of Rakesh Aggarwal of UGS. This paper was funded by Deere & Co. We are also thankful to the Virtual Reality Applications Center for the use of computational resources and virtual reality hardware.

References

[1] Boothroyd, G., and Dewhurst, P., 1989, *Product Design for Assembly*, McGraw-Hill, New York.

[2] Seimens Vis Jack, 2010, https://www.plm.automation.siemens.com/en_us/products/tecnomatix/assembly_planning/jack/vis_jack.shtm

[3] Fröhlich, B., Tramberend, H., Beers, A., Agarawala, M., and Baraff, D., 2000, "Physically-Based Modeling on the Responsive Workbench," Proceedings of the IEEE Virtual Reality Conference, New Brunswick, NJ.

[4] Seth, A., Su, H. J., and Vance, J. M., 2006, "SHARP: A System for Haptic Assembly and Realistic Prototyping," ASME Paper No. DETC2006/CIE-99476.

[5] Dewar, R. G., Carpenter, I. D., Ritchie, J. M., and Simmons, J. E. L., 1997, "Assembly Planning in a Virtual Environment," Proceedings of the Portland International Conference on Management of Engineering and Technology (PICMET 97), Portland, OR.

[6] Ritchie, J. M., Dewar, R. G., and Simmons, J. E. L., 1999, "The Generation and Practical Use of Plans for Manual Assembly Using Immersive Virtual Reality," Proc. Inst. Mech. Eng., Part B, **213**(5), pp. 461–474.

[7] Ritchie, J. M., Simmons, J. E. L., Carpenter, I. D., and Dewar, R. G., 1995, "Using Virtual Reality for Knowledge Elicitation in a Mechanical Assembly Planning Environment," Proceedings of the 12th Conference of the Irish Manufacturing Committee.

[8] Gupta, R., and Zeltzer, D., 1995, "Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments," Proceedings of the ASME Computers in Engineering Conference and the Engineering Database Symposium, Boston, MA.

[9] Gupta, R., Whitney, D., and Zeltzer, D., 1997, "Prototyping and Design for Assembly Analysis Using Multimodal Virtual Environments," Comput.-Aided Des., **29**(8), pp. 585–597 (Special issue on VR in CAD).

[10] Coutee, A. S., McDermott, S. D., and Bras, B., 2001, "A Haptic Assembly and Disassembly Simulation Environment and Associated Computational Load Optimization Techniques," ASME J. Comput. Inf. Sci. Eng., **1**(2), pp. 113–122.

[11] Coutee, A. S., and Bras, B., 2002, "Collision Detection for Virtual Objects in a Haptic Assembly and Disassembly Simulation Environment," ASME Paper No. DETC2002/CIE-34385.

[12] Baraff, D., 1995, "Interactive Simulation of Solid Rigid Bodies," IEEE Comput. Graphics Appl., **15**(3), pp. 63–75.

[13] Krüger, W., and Fröhlich, B., 1994, "The Responsive Workbench," IEEE Comput. Graphics Appl., **14**(3), pp. 12–15.

[14] Mandiak, M. A., and Kesavadas, T., 2005, "Development of Virtual Assembly Application With Haptics, Assembly Modification and Statistical Output Measures," ASME Paper No. IMECE2005-82783.

[15] Whitney, D. E., 2004, *Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development*, Oxford University Press, New York.

[16] Kim, C. E., and Vance, J. M., 2003, "Using VPS (Voxmap Pointshell) as the Basis For Interaction in a Virtual Assembly Environment," ASME Paper No. DETC2003/CIE-48297.

[17] Kim, C. E., and Vance, J. M., 2004, "Collision Detection and Part Interaction Modeling to Facilitate Immersive Virtual Assembly Methods," ASME J. Comput. Inf. Sci. Eng., **4**(2), pp. 83–90.

[18] McNeely, W. A., Puterbaugh, K. D., and Troy, J. J., 1999, "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling," Proceedings of the SIGGRAPH 99 Conference, Annual Conference Series, Los Angeles, CA.

[19] Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., and Hart, P., 1999, "VADE: A Virtual Assembly Design Environment," IEEE Comput. Graphics Appl., **19**(6), pp. 44–50.

[20] Jayaram, S., Connacher, H. I., and Lyons, K. W., 1997, "Virtual Assembly Using Virtual Reality Techniques," Comput.-Aided Des., **29**(8), pp. 575–584.

[21] Jayaram, U., Tirumali, H., and Jayaram, S., 2000, "A Part/Human Interaction

Model for Assembly in Virtual Environments," ASME Paper No. DETC/CIE-14584.
[22] Taylor, F., Jayaram, S., and Jayaram, U., 2000, "Functionality to Facilitate Assembly of Heavy Machines in a Virtual Environment," ASME Paper No. DETC/CIE-14590.
[23] Jayaram, S., Jayaram, U., Wang, Y., and Lyons, K., 2000, "CORBA-Based Collaboration in a Virtual Assembly Design Environment," ASME DETC 2000/CIE-14585.
[24] Wang, Y., Jayaram, S., Jayaram, U., and Lyons, K., 2001, "Physically Based Modeling in Virtual Assembly," ASME Paper No. DETC2001/CIE-21259.
[25] Marcelino, L., Murray, N., and Fernando, T., 2003, "A Constraint Manager to Support Virtual Maintainability," Comput. Graph., **27**(1), pp. 19–26.
[26] Wan, H., Gao, S., Peng, Q., Dai, G., and Zhang, F., 2004, "MIVAS: A Multi-Modal Immersive Virtual Assembly System," ASME Paper No. DETC 2004/CIE-57660.
[27] Liu, Z., and Tan, J., 2005, "Virtual Assembly and Tolerance Analysis for Collaborative Design," Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design, Coventry, UK.
[28] Chen, X., Xu, N., and Li, Y., 2005, "A Virtual Environment for Collaborative Assembly," Proceedings of the Second International Conference on Embedded Software and Systems (ICESS '05).
[29] Howard, B. M., and Vance, J. M., 2007, "Desktop Haptic Virtual Assembly Using Physically Based Modeling," Virtual Real., **11**(4), pp. 207–215.
[30] Coutee, A. S., and Bras, B., 2004, "A Comparison of Two Collision Detection Libraries in a Haptic Simulated Environment," ASME Paper No. DETC2004-57673.
[31] Seth, A., Su, H. J., and Vance, J. M., 2005, "A Desktop Networked Haptic VR Interface for Mechanical Assembly," ASME Paper No. IMECE2005-81873.
[32] Seth, A., Su, H. J., and Vance, J. M., 2008, "Development of a Dual-Handed Haptic Assembly System: SHARP," ASME J. Comput. Inf. Sci. Eng., **8**(4), p. 044502.
[33] Just, C., Bierbaum, A., Baker, A., and Cruz-Neira, C., 1998, "VR Juggler: A Framework for Virtual Reality Development," Proceedings of the Second Immersive Projection Technology Workshop, Ames, IA.
[34] Hoffmann, C. H., 2001, "D-Cubed's Dimensional Constraint Manager," ASME J. Comput. Inf. Sci. Eng., **1**(1), pp. 100–101.